The DII COE: Basic Principles and Future Challenges

Doug Gardner
Defense Information Systems Agency

While not a silver bullet, the Defense Information Infrastructure (DII) Common Operating Environment (COE) is a measured, pragmatic approach to software development and integration that is tailored to the Department of Defense. The four basic principles of COE – interoperability, security, customer focus, and best value – are what drives the COE as it responds to past successes and seeks to embrace evolving open frameworks and object technology. This article entails a description of how these principles are addressed in the COE development process. This is critical to understanding the goals of the COE environment: How it is used. How it should be used. Also discussed are the challenges facing the COE as a result of decreasing commercial product life cycles, rising customer expectations, and increased demand to support new communities.

The Defense Information Infrastructure (DII) Common Operating Environment (COE) was created to provide the Department of Defense (DoD) with software processes and products that accommodate the unique corporate characteristics of the DoD. Since its inception, the COE has been embraced by all the major service Command and Control systems and the intelligence community as the basis for combined joint interoperability from the National Command Authority through the commanders-in-chief to the joint task force.

As embodied in the Global Command and Control System (GCCS) and its service variants, COE has been employed in all major theater operations. These include the U.S. Central Command operations SOUTHERN WATCH (no fly-zone enforcement) and DETERMINED RESPONSE (USS COLE aftermath), U.S. European Command operations DELIBERATE FORGE (NATO Air Operations), JOINT FORGE (Stabilization Forces), and SILENT PROMISE (South African relief).

As COE embraces evolving open frameworks and object technology, such as the Web, publish-and-subscribe data sources, portals, and component software, hundreds of new system integrators have moved to the COE.

What drives the COE as it responds to these past successes and seeks to support a broader application base? The answer to this question is encompassed in the COE's four basic principles: *interoperability*, *security, customer focus*, and *best value*. A description of how these principles are addressed in the COE development process is critical to understanding the goals of the COE environment, how it is built, and how it should be used.

Interoperability

Interoperability of joint systems is critical to success on the modern battlefield. However, interoperability is about making limiting choices. The DoD has tried many initiatives to make joint systems interoperable, most of them based on developing and mandating standards (such as Ada, POSIX[1], and the Joint Technical Architecture). While these efforts have had some success, the COE goes beyond interoperability through standards and provides interoperability through products. Common products make for common software behavior and reduce the number of avenues that developers can use to move away from a common, interoperable implementation.

> *"The COE, as the primary vehicle for providing controlled infusion of new technology across a large number of systems, is squarely in the crosshairs of the expectations gap."*

The DII COE is based on the thesis that having the exact same software on both sides of an interface is the most effective way to ensure consistent behavior across the interface. Thus interoperability between systems can be measured by how much of the exact same software is shared between the systems. Interoperability in the COE is a spectrum: its minimal level of interoperability being federation (the ability to run two applications on the same platform without stepping on each other), and the maximum level of interoperability being two systems that are entirely identical except for domain or functionally specific mission applications.

There are some practical advantages to high levels of interoperability as defined by the COE, some of which are particularly important in the DoD environment. For example, we expect that in future contingencies a military user will need to run a functionally specific application on a system supplied by another service (e.g., the Army logistician who needs to run an Army application while assigned aboard a Navy ship as part of a joint task force). High levels of interoperability, as defined by the COE, will make this easier by ensuring that the underlying infrastructure of both the originating Army system and the receiving Navy system use the same software.

It is a continual challenge for the COE to balance the need to build common software with the legitimate requirements of system developers for specialized or unique services. In each of these cases, the potential for providing services in the COE that could reduce interoperability (for example, multiple products that provide the same service) must be weighed against the specialized requirement. This approach is designed to help the community make limiting choices together, and while it has resulted in a substantial amount of community agreement, it is an approach that is not very popular with developers and users who are accustomed to having complete control of functional and system design decisions.

A common statement from a program exploring using the COE is, "I can't use the COE because it doesn't have or do _____ (fill in the blank)." If the COE can be expanded to handle that blank space, then every effort is made to address the issue in the COE software baseline. If the request is incompatible or would introduce new opportunities for COE customers to make conflicting decisions, then the requested capability is not

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2001		**00-00-2001 to 00-00-2001**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The DII COE: Basic Principles and Future Challenges	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Information Systems Agency,Arlington,VA,22204**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
CROSSTALK The Journal of Defense Software Engineering, October 2001

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**5**	

included in the COE. If the program then chooses to not use the COE, the program should clearly understand that they have just prioritized their missing desire above joint interoperability, or at least the level of joint interoperability you get from the COE. In some cases, this is an understandable trade-off for a program to make (their need really is more important than the gains afforded by the COE). However, in most cases, the decision to not use the COE will make the integration of that program's software into a joint system considerably more difficult.

Security

Security is an important factor in the COE, and as might be expected, has been growing in importance over the past few years. Since the COE is not a system, it does not go through the formal security accreditation required of DoD systems. Instead, components being included in the COE are assessed against a stringent set of security guidelines, and are occasionally rejected if their acceptance would create an unacceptable level of risk in end systems. The COE provides recommendations, in the form of default software configuration settings, for how COE customer systems should use COE components to maximize security. Ultimately, however, and this is a very important point, the security posture and level of acceptable risk are the decision of the developers of the systems that use the COE.

Aside from the secure components and configuration guidance, the COE provides additional security-related services to developers of COE-based systems and applications. First, by using the COE, a customer system inherits a basic amount of security that serves as a community-wide, common level of reasonable basic security. This basic security serves as a starting point for developers and allows them to increase the security of their systems to meet specialized security requirements.

The second primary security service provided as part of the COE process is a watchdog organization (the COE Security Team) that continuously monitors threats to COE software, informs the community of potential concerns, and ensures that appropriate patches or configuration advice are made available as quickly as possible. The COE Security Team monitors DoD security initiatives (such as Information Alert Vulnerability Assessments), as well as security publications and commercial product announcements, to identify security issues that are relevant to the COE customer community. This focus on security issues associated with COE components is designed to reduce the amount of vulnerability research required by COE-based system developers and administrators.

A third focus of the COE security support effort is to provide tools that system integrators and application developers can use to enhance the security of their products. One tool is a configuration-based tool that examines a machine and identifies potential security vulnerabilities such as file permissions on UNIX that are too permissive or scripting conventions that can be readily exploited. Another tool is a set of interfaces that allow application and system developers to embed secure authentication and data transport into their software. These tools can be used by developers to build security into systems, and they will make it easier to build systems with consistent security implementations. The tools will be expanded over time to support new security mechanisms and services.

"This component-based view of system development encourages system integrators to meet system requirements by looking for existing components ..."

Customer Focus

Ultimately, the DII COE is about building systems. In the DoD "system-building universe," there are four primary COE customers: system integrators, application developers, system administrators, and functional users. This diverse set of customers has very different requirements. The COE's success depends on our ability to balance the capabilities provided in the COE across these four groups.

System Integrators

System integrators are those DoD development organizations that are responsible for providing integrated, full-service system solutions that include both general purpose capabilities (such as office automation and Web browsers), as well as functionally specific capabilities (such as situational awareness, transportation management, or financial services). For the military command and control community, the primary systems that use the COE are the Global Command and Control System (GCCS), GCCS-Maritime, GCCS-Army, Theater Battle Management Core Systems, and the Army Battle Command System.

System integrators rely on the DII COE to provide a common integration methodology that supports widespread sharing of applications across systems and ensures that applications developed according to COE guidelines will peacefully coexist in an integrated system. The COE allows system integrators to view the integrated system as a series of components that can be packaged together or flexibly deployed to meet specific needs of system users.

This component-based view of system development encourages system integrators to meet system requirements by looking for existing components, either provided by the COE or built by another COE-compliant developer in the DoD. This leveraging of other development efforts allows the system integrator to focus resources on the new or functionally unique portions of the system rather than on general purpose tools and components.

The COE has two key rules designed specifically to support system integrators: software services provided by the COE are not retired except through a process of community agreement, and services in the COE are upgraded with an eye toward full backward compatibility. This ensures that applications built using the old services continue to run in the upgraded environment without any modification. These goals allow system integrators to use COE services or COE-based applications knowing that services in the COE will not disappear without warning, and that all customers will have a say in the retirement schedule. For a large system that includes hundreds of functional components, these guarantees allow the system integrator considerable flexibility in upgrading individual components of the system over time, instead of having to reengineer all capabilities before any new infrastructure features can be fielded.

System integrators can choose to include in their systems software developed by the COE. The COE provides the standard implementation for key command and control functionality, specifically the data management, dissemination,

and visualization components that make up the Common Operational Picture (COP). The COP is a framework and a set of standard capabilities built against that framework that provide a common set of data, an integrated display capability, and a robust data replication mechanism in support of situational awareness across the joint community. Also, the COE provides a set of government-built services that are not supported by commercial products (such as profiles, which extend a user's account definition to support shift work and multiple duties, and cross-platform account management, which includes a single tool for creating accounts that work across all COE-supported platforms).

Application Developers

Application developers are those DoD development organizations that build software to address a particular functional area such as weather, intelligence, or logistics. While applications are usually built to be part of a particular system, some are built to be fielded as components of multiple systems. Two current examples of joint mission applications are the Integrated Imagery and Intelligence application suite and the Air Tasking Order Exchange capability.

Primarily, the COE supports application developers by allowing them to focus their resources on developing mission-specific functionality instead of infrastructure. Before the acceptance of the COE, functional applications were usually not built to be part of a larger system. Each application required its own account management tools, map package, data management engine, etc. For many applications, these components could eat up well over half of the development budget without ever providing the end user any specific functionality. By using the COE, the money allocated to develop infrastructure components can instead be reprogrammed toward meeting more of the end user's functional requirements.

In addition, reliance on COE infrastructure components also helps the application developer ensure that the application can be integrated into multiple systems with minimal effort. For example, community acceptance of a standard mapping package means that separate versions of the application for different map engines are not required. The application developer has a clear idea of the target environment during development and therefore can make appropriate decisions without a great deal of coordination.

The COE also provides application developers with a public clearinghouse for requirement and product information. The DII COE has chartered 19 technically focused working groups to document requirements, incorporate technical advice, and recommend products for inclusion in the COE. These working groups conduct a significant amount of research, both in terms of what is commercially available and how well various products fulfill community requirements. These are public forums that include experts from across the DoD community. Participation in these groups, or at least monitoring their progress, is a good way to reduce the amount of resources that a development organization has to spend to research both industry and the rest of the DoD.

" ... the COE supports application developers by allowing them to focus their resources on developing mission-specific functionality instead of infrastructure. "

System Administrators

System administrators are the people tasked with making DoD systems run in the field. They are usually military specialists and are almost never the same people who develop the applications or integrate the systems to be operated and maintained. To do their job properly, they need clear system documentation, automated installation and upgrade tools, and predictable system environments.

The COE supports system administrators by forcing application developers and system integrators to consider, as part of the development process, the impact of design decisions on the workforce that will make the systems run in the field. As part of the overall COE philosophy, it attempts to minimize the effort required to maintain and upgrade COE-based systems while providing enough flexibility to allow operational sites to adequately control their local information technology resources (e.g., to install the software they need to accomplish their mission).

The COE tool supporting these goals is the COE Installer, which provides a point-and-click interface for loading, updating, and configuring software on COE-based systems. The COE Installer performs specialty features like installations from a network server, dependency checking to ensure required components are loaded in the proper order, and version checking to ensure that applications have access to the proper versions of commercial and government-built infrastructure components. COE rules about separate directories for each component ensure a predictable "laydown" of software on each machine, and that the installation of a new component will not remove or overwrite software required by an already loaded component.

Functional Users

Functional users are the military operators who use the systems built on the COE. They use their computer systems to perform practically all aspects of their day-to-day duties, as well as to prosecute joint military operations. Systems built on the COE support users across the spectrum of military operations from routine administrative users to those conducting operational planning and execution. COE-based systems also provide support to users at all echelons of the command structure hierarchy (from the National Command Authority to the foxhole).

The functional users are the ultimate customers of the COE, but what the COE provides is largely invisible to them. Although concepts such as component installation, system integration, and software resource management are outside the functional users' scope (and interest), these concepts provide the processes and software framework for joint cooperation and integration that make it possible to execute modern joint operations. The COE provides the mechanism to bring together applications from a variety of systems, quickly and under less than ideal conditions, to automate a joint task force. One immediate pay off of the COE in this area is that it has almost completely ended the practice of each new functional capability being delivered to the end user as a new system with new hardware, training, and administration requirements. New functionality is now commonly delivered into a system, so users do not have to manage multiple workstations each providing only a portion of the necessary information in order to do their jobs. Systems like GCCS, and their service counterparts, now serve as the host systems for new functionality

being offered to users.

Another key benefit to the functional users is that, for the same amount of funding across the DoD, the COE allows less to be spent building duplicative (and often non-interoperable) infrastructure components. This frees up more money to be focused on the functional requirements that directly support mission accomplishment. This redirection of resources is already starting to become evident in some functional domains within the DoD and is likely to become more obvious over the next few years as current service and joint systems upgrade their infrastructure to the latest version of the COE.

In the area of the COP, the COE provides functional users a situational awareness capability that can expand and change as new sources, sensors, and decision tools become available. The challenge from the COE perspective is to make dozens of COP applications and decision tools, built by a wide variety of government and commercial organizations, appear to the end user as if they were built as part of a single, integrated suite of applications. By providing just such a framework, the COP has become the primary integration mechanism for command, control, and intelligence data across the DoD. It will continue to evolve to allow the broadest amount of flexibility for contributing applications to bring additional information to the common display and to facilitate the secure distribution of data needed by decision-makers at all levels.

Best Value

Although not the primary focus of the COE, an extremely important byproduct of achieving system and application interoperability through common software is cost savings. The key aspect of cost savings in the COE is the use of commercial products, which make up about 85 percent of the COE. In almost every case, COTS products are less expensive for the government to acquire, modify, and enhance than government-built components.

As for the government-built products in the COE, they are developed either because available COTS products would make the COE less interoperable or because the required functionality does not exist in any commercial product. Although it seems axiomatic that using an existing government-built product rather than building your own would save money, the amount of money saved has been difficult to quantify. This is probably because most programs don't track cost savings, but the fact is that the DoD has not invested the time and resources required to quantify costs attributable to the COE.

However, there are areas where savings due to community-wide adoption of the COE can be quantified. A particularly good example is the Integrated Imagery and Intelligence set of mission application capabilities built by the Navy and being fielded on each of the COE-based major service command and control systems. Instead of four different development efforts, the DoD will pay for only one. As this model is applied to each of dozens of functional areas across the military, this aspect of cost savings will likely represent the biggest financial advantage of the COE for the DoD.

"The COE, as the primary vehicle for providing controlled infusion of new technology across a large number of systems, is squarely in the crosshairs of the expectations gap."

However, there are packaging, and at higher levels of COE compliance, reengineering costs associated with migrating to the COE. The initial cost to move to minimal COE compliance is usually very low, partly because the COE was designed that way and partly because minimal COE compliance is based on adhering to generally recognized good software development practices. Depending on how tightly an application is integrated with its infrastructure, achieving higher levels of COE compliance can incur moderate to high costs. The savings in long-term maintenance balance some of these migration costs, but mostly the costs for migration to the COE should be viewed as the cost of becoming joint.

One other cost-saving outcome of the COE's role as a repository for common capabilities is that the COE has become a single forum that represents service and agency software requirements to industry. This allows the COE (usually through technical working and advisory groups) to represent a broad range of the DoD in discussions with industry. It also allows the COE to arrange COE-wide licenses for certain key capabilities, such as printing and Web services.

Future Challenges

The COE faces many challenges in the future, particularly in the areas of keeping up with technology, maintaining a collaborative atmosphere with our customers, balancing the use of commercial products and services with the need to maintain open software solutions, and expanding the COE to take advantage of other services and technologies.

The "Expectations Gap;" the Treadmill Keeps Getting Faster

The most significant challenge for DoD software development in general and the DII COE in particular is the growing mismatch between the amount of time it takes to field a system and how quickly commercial industry is moving. The COE initially assumed service systems would upgrade their software and hardware infrastructure every three years. The reality, however, is that the current versions of fielded systems will not be upgraded for five to seven years, for a variety of reasons: specialized security requirements, in-depth functional testing, expense of retraining users and system administrators, operational concepts that lag technological innovation, and scheduling availability for operationally deployed forces, just to name a few.

So what DoD software developers face is a growing expectations gap by users who see new capabilities in the commercial marketplace that are still years away from being systematically deployed in the DoD. The COE, as the primary vehicle for providing controlled infusion of new technology across a large number of systems, is squarely in the crosshairs of the expectations gap. We need to move fast enough to keep up with the lightning pace of industry while not leaving any legacy systems or applications behind.

A factor that will increase the expectations gap is the frequency with which commercial products are being replaced and/or retired by their manufacturers. New and improved products are being produced much faster than large-scale system developers can keep up. With each new release, the commercial business mind-set is that an older product becomes *unsupported* (both to save the manufacturer on the number of baselines to maintain and to *encourage* customers who haven't upgraded in a while to move to the later

version). For some commercial products, the release-to-retirement cycle fits inside the typical DoD system "develop, test, and field cycle." That means that in the time between code freeze for validation, certification, training, and fielding, some of the commercial products in the *frozen* baseline are becoming unsupported by vendors. While the COE guarantees that its government-built interfaces will remain backward compatible and fully supported, it is not possible to make the same claim for commercial products.

This Only Works if Everyone Works Together

The DoD consists of hundreds of autonomous, decentralized software development and acquisition organizations, each of which contributes a portion of the overall capability required to prosecute joint operations. With so many agendas and specific needs that ultimately are required to come together to support the decisions of a single commander, it is critical that there be an open dialogue to reconcile the conflicting demands of systems contributing to joint operations. The COE provides a forum for discussing the technical tradeoffs associated with joint software development. It can continue to be a useful part of the overall joint solution if the services and agencies that participate continue to make being joint a priority.

The Not-So-Hidden Threat From Industry

The DoD is a highly competitive arena, at times within the government but certainly among the defense contractors and commercial vendors who provide government software services.

From the standpoint of the commercial marketplace, the COE conflicts with the corporate agendas of most commercial vendors. The common development and integration approach of the COE discourages any program to become dependent on a proprietary approach offered by a particular vendor. The COE enforces an anti-monopoly stance by providing services in such areas as cross-platform support and by including multiple commercial products where doing so allows systems to make cost/feature tradeoffs without sacrificing interoperability.

A related challenge for the DII COE is trying to balance the DoD's desire to use commercial products with corporate business models that push for product uniqueness and proprietary approaches. A perfect example of this is the DoD goal of cross-platform consistency. The ideal is that platforms in the DoD should be interchangeable, that is, they should provide a common set of services invoked the same way. This would allow applications and software tools to be more readily shared across systems, thus saving the DoD millions (perhaps billions) of dollars and ensuring consistent behavior for all users. Instead, commercial industry spends billions to ensure that the ideal is never reached – there is no business case for making a product the same as a competitor's. Even where standards exist, such as Structured Query Language for relational databases, the *extensions* provided by each database vendor ensure that applications built for one product cannot be moved to another product without significant reengineering.

Controlled Growth for the COE

There is considerable pressure on the COE to expand to support new technical requirements, particularly in the areas of real time and tactical systems support.

Expansion into the real-time environment will require support for a much larger set of hardware and operating system configurations. It will also require fundamental reengineering of some COE applications to both adhere to more stringent processing requirements and to take advantage of new services provided by real-time operating systems. The differences in system development and integration philosophies between the real-time and non-real-time communities have already challenged some of the core COE concepts. There is a clear need for the COE to provide the tools and products that allow integration between the decision-making systems that support a joint task force and their real-time counterparts. This will be a significant focus area in future deliveries of the COE.

Providing support to the tactical community will challenge the COE to operate on smaller hardware and to support future developments in wireless technologies, hand-held Personal Digital Assistants, and radios. More flexibility in the amount of bandwidth used, more control over the flow of data, and more visualization options will be required. The requirements in this space are just beginning to be defined, but this is also clearly an area that the COE will need to support in future deliveries.

The challenge to the COE will be to incorporate these capabilities while maintaining a stable baseline for current COE customer systems that are in the field.

Conclusion

The DII COE is a measured, pragmatic approach to software development and integration that is tailored to the DoD. It is a customer-driven and cost-conscious process that results in products that are interoperable and secure. The challenges facing the COE over the next few years are significant as the trends of decreasing commercial product life cycles, rising customer expectations, and increased demand to support new communities converge. As the COE evolves in the future, it is critical to understand that it is not a silver bullet. Successful software development in the DoD still requires good systems engineering, disciplined development processes, and detailed coordination among related applications and systems.◆

Note

1. POSIX is a registered trademark of The Institute of Electrical and Electronic Engineers, Inc. (IEEE).

About the Author

Doug Gardner has worked on practically every part of the Defense Information Infrastructure (DII) Common Operating Environment (COE) since he began with the Defense Information Systems Agency (DISA) in 1996. After serving as the Common Support Applications Team chief for two years, he was named as the COE chief engineer in April 2001. He has a master's degree in defense policy from The Claremont Graduate School and a bachelor's degree in electrical engineering and computer science from Rice University. Prior to coming to DISA, he worked for the Jet Propulsion Laboratory as a software developer on a variety of command and control, intelligence, and modeling and simulation systems. Gardner is also a major in the U.S. Army Reserves.

Defense Information Systems Agency
Phone: (703) 681-2328
E-mail: gardnerd@ncr.disa.mil